Statistical Approach For Detection Of Vehicle In Heavy Traffic

Rupali Patra

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela-769008, Odisha, India

Statistical Approach For Detection Of Vehicle In Heavy Traffic

Thesis submitted in partial fulfillment of the requirements for the degree of

Bachelor of Technology

in

Electronics and Communication Engineering

By

Rupali Patra

(110EC0422)

Under the guidance of

Prof. L. P Roy

Department of Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela -769008, India

Electronics and Communication Engineering

National Institute of Technology Rourkela

Rourkela-769008, India.

Certificate

This is to certify that the thesis titled, “Statistical approach for Detection Of Vehicle In Heavy Traffic” submitted by Rupali patra Bearing Roll no 110EC0422 in partial fulfillment of the requirements for the award of Bachelor of Technology Degree in Electronics and Communication Engineering at National Institute of Technology, Rourkela (Deemed University) in an authentic work carried out by her under my supervision and guidance.

Prof L P Roy

Dept. of Electronics and Communication Engg.National Institute of Technology, Rourkela-769008

Acknowledgment

This dissertation, though an individual work, has benefited in various ways from several people. Whilst it would be simple to name them all, it would not be easy to thank them enough .I would like to express my sincere thanks to Prof.L P Roy sir for his constant guidance and support throughout the course of project. Overwhelming thanks to all members of the Department of Electronics and communication Engineering, NIT Rourkela for their encouragement and co-operations throughout the course of project .I would like to extend my gratitude to all the researchers and scholars whose papers and thesis have been utilized in the project. Finally, I dedicate my thesis to my parents for their love, support and encouragement without which this would not have been possible.

RUPALI PATRA

Contents

ABSTRACT

Today Security is given very much importance and lot of electronic equipment is being used in security applications. Monitoring continuously the movements of persons or vehicles and reporting when predefined events take place is very a common application .These systems are being used to prevent terrorism, accidents ,traffic congestion and efficient monitoring of traffic. A human observation based system for implementing this has several disadvantages. The present day technology allows automatic detection based on predefined measures. In this thesis an innovative system for detecting and extracting vehicles in traffic surveillance scenes is presented .The main concept behind vehicle detection in a live video is extract the foreground and remove the background from it .This theory is called background subtraction .This method can be implemented in various ways such as setting a particular threshold value and removing the objects having value less than it .The second approach is to compare the current frame with the previous frame and if the variance is more than a certain value it detects the motion of that object .Third and the most efficient method is to use a statistical method where a certain number of video frames are used to initialize a fixed number of Gaussian modes in the mixture model. While in the first method only white cars are being detected this disadvantage is solved when we use a statistical method where a particular vehicle is detected using a foreground detection technique on a frame .Here the input video file is read in AVI format .After that morphological operations are done on it and the bounding box is calculated .Finally the moving object is presented with a rectangle drawn around it and total number of vehicles in the current frame is calculated .This process is repeated for each frame till the whole video is processed .Since this method uses a training set and not a general threshold selected manually by the user the foreground extracted is more desirable than other method and besides it requires much less memory than the method where the background subtraction is done by comparing the frame with the previous one .And last but not the least it gives a general idea about the vehicle frequency in the video which can be very helpful in traffic monitoring.

1 .INTRODUCTION

As urban road intersections are prone to traffic congestion and traffic accidents, monitoring the crossing of vehicles and predicting the state is needed to reduce traffic congestion and prevent accidents. Background subtraction concept is used to track the vehicles .Vehicle tracking is done for driving behavior identification and traffic incident detection. Conventional technology for traffic measurements, such as sonar or microwave detectors, inductive loops suffer from some serious drawbacks: they are expensive to install, they demand traffic disruption during installation or maintenance, they are not portable and they are unable to detect slow or stationary vehicles. On the contrary, video based systems are easy to install, they can be easily upgraded and they offer the flexibility to redesign the system and its functionality by simply changing the system algorithms. Those systems allow vehicle counting, classification, measurement of vehicle's speed and the identification of traffic incident .A typical video surveillance system has to face many challenges such as:

1- vehicles vary in size,shape ,colour
2- changing lighting and weather conditions
3- occlusion of a vehicle by another one etc.

A typical background subtraction detects the actual background and extracts objects that do not belong to it. The concept of this method is in a typical background model a prototype of the image background (an initialization of the background) is considered first and then each pixel of the prototype is compared with the actual image color map. If the color difference exceeds a predefined threshold it is assumed that this pixel belongs to the foreground .Otherwise it is considered as the background.The efficiency and performances of the monitoring systems lies on the algorithm used for background subtraction and vehicle detection.That is how fast it works ,how much memory it needs and how efficiently it detects the vehicles.

1.1**Background subtraction**:

Background subtraction is simply separating the foreground(object of intrest) from the background .This can be carried out in various ways .A video is first converted into separate frames .Then each frame is compared with the previous frame .Since the camera is still the background would remain constant .Hence change or motion is detected if the variance crosses a certain threshold value. And this changed part is the moving object .Second method is setting a general threshold value for the whole video .The values that lie above it are considered as the foreground and rest all part are rejected .Since a general threshold value is selected there is possibility of presence of other unwanted noise such as line markings on the road etc .To remove these morphological operations are done on it .Third and the most efficient way of doing the background subtraction is providing a statistical method a set of input training frames .This method will then calculate the mean and variance of the frames .After that it detects the foreground based on the values that cross the threshold value by a specific variance .

1.2 Vehicle detection:

After the background subtraction is done on the frames the remaining objects has to pass through a some morphological operations to get cleaner foreground .This step is necessary since there is a he possibility of presence of unwanted objects in the foreground along with some additional noise .After a vehicle is detected it is marked in a certain way such as marking it with a mark on it or bounding it with a rectangle .The next is to calculate the number of the vehicles based on the output and average frequency of the vehicles .During the frame differencing the following points should be taken into consideration:

• **Motion in the background**: Non-stationary background regions, such as branches and leaves of trees, a flag waving in the wind or a towed car etc should be identified as part of the background.

• **Illumination changes**: The background model should be able to adapt to changes in the illumination from day time to night time and it includes the weather conditions

• **Memory**: The background module should not use much resource, in terms of computing power and memory.

• **Shadows**: Shadows cast by moving object should be identified as part of the background and not foreground.

• **Camouflage**: Moving object should be detected even if pixel characteristics are similar to those of the background.

2.Gaussian mixture Model

Definition

A Gaussian Mixture Model (GMM) is a parametric probability density function represented as the sum of Gaussian component densities. GMMs are used as a parametric model of the These parameters are estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum *A Posteriori* (MAP) estimation from a well-trained preceding model.

Introduction:

A Gaussian mixture model is a weighted sum of n component Gaussian densities and can be represented by the equation,

$$P(x|\mu)=\sum_{i=1}^{n}(wi)g(x|\mu)\sum i$$

where x is a D-dimensional continuous-valued data vector (i.e. measurement or features), wi, i = 1, . . . ,M, are the mixture

weights, and g(x|μi,_i), i = 1, . . . ,n, are the component Gaussian densities. Each component density is a D-variate Gaussian function of exponential form.

Here μi is mean vector and _i is the covariance matrix. The mixture weights satisfy the constraint that i=1 wi = 1.

The complete Gaussian mixture model is parameterized by the mean vectors, mixture weights and covariance matrices from all component densities.

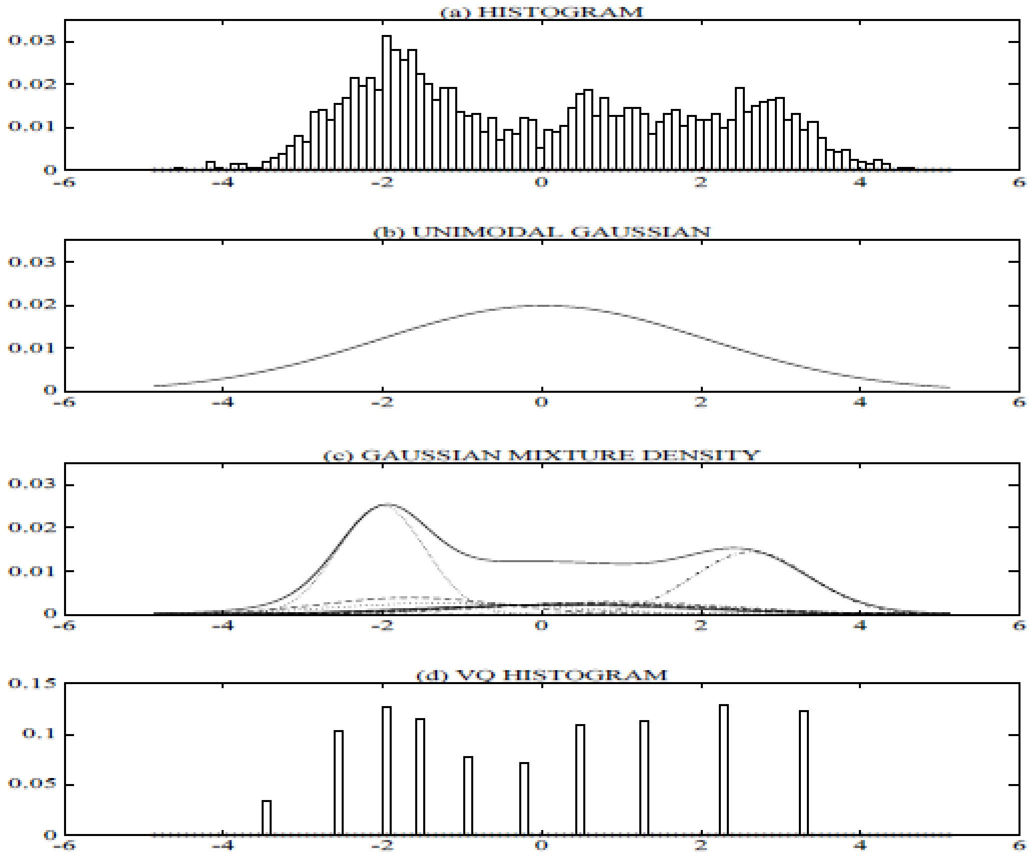
As the component Gaussian are acting together to model the overall feature density,full covariance matrices are not necessary even if the features are not statistically independent. The linear combination of covariance basis Gaussians is capable of modeling the correlations between the feature vector elements. The effect of using a set of n covariance matrix Gaussians can be equally obtained by using a larger set of diagonal covariance Gaussians.

GMM is able to make smooth approximations to arbitrarily shaped densities.

The uni-modal Gaussian model represents feature distributions by a position (mean vector) and a elliptic shape (covariance matrix) and a nearest neighbor model represents a distribution by a discrete set of characteristic template. A GMM acts as a hybrid between these two models

by using a discrete set of Gaussian functions, each with their own mean and covariance matrix, to allow a better modeling capability. Figure 1 compares the densities obtained using a unimodal

Gaussian model, a GMM and a VQ model. Plot (a) displays the histogram of a single feature from a speaker recognition system ,plot (b) shows a uni-modal Gaussian model of this feature distribution; plot (c) shows a GMM and its 10 underlying component densities; and plot (d) shows a histogram of the data assigned to the VQ centroid locations of a 10 element codebook. The GMM not only provides a smooth overall distribution fit, its components also clearly detail the multi-modal nature of the density.



0

With the given set of training frames and configuration we calculate the GMM parameters using maximum likelihood operations.For a set of T training frames the likelihood can be represented as

$$p(X|\lambda) = \prod_{t=1}^{T} p(\mathbf{x}_t|\lambda).$$

The basic idea of this model is use a set of models to estimate a new set of models of same size.

This is done using the mean,variance and the mixture weights.

Mixture weights

$$\bar{w}_i = \frac{1}{T}\sum_{t=1}^{T} \Pr(i|\mathbf{x}_t, \lambda).$$

Mean

$$\bar{\boldsymbol{\mu}}_i = \frac{\sum_{t=1}^{T} \Pr(i|\mathbf{x}_t, \lambda)\, \mathbf{x}_t}{\sum_{t=1}^{T} \Pr(i|\mathbf{x}_t, \lambda)}.$$

Variance

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^{T} \Pr(i|\mathbf{x}_t, \lambda)\, x_t^2}{\sum_{t=1}^{T} \Pr(i|\mathbf{x}_t, \lambda)} - \bar{\mu_i}^2,$$

Where σ_i^2, x_t, and μ_i hold their usual meanings.

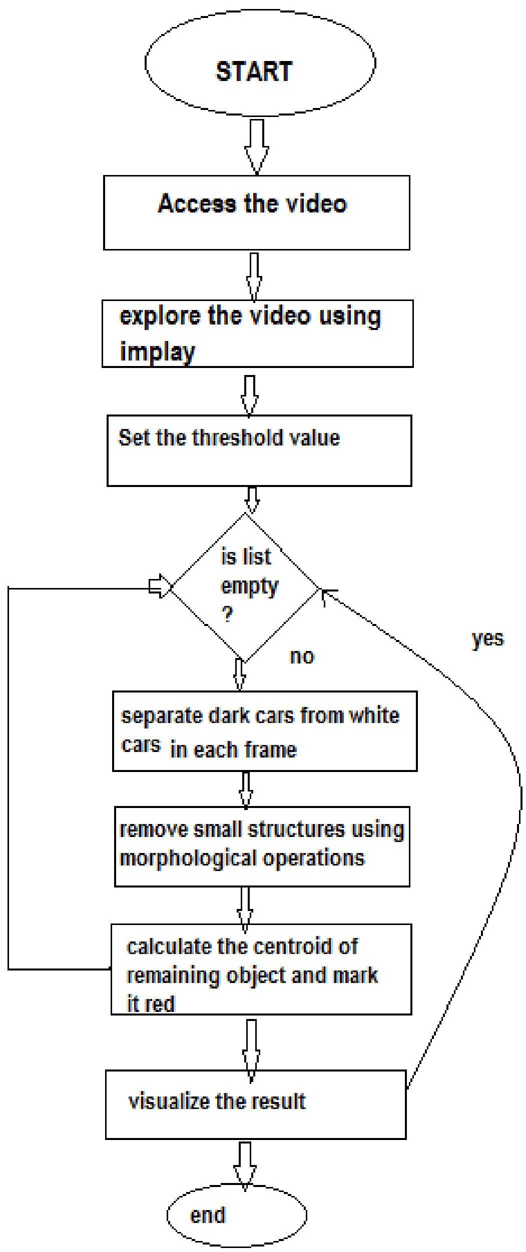
Since these parameters are calculated using the training input set hence this model gives a more desirable set of output .As a result foreground detected using these parameters are more accurate than other methods.

3 .Algorithms implemented in the project

In the current thesis we have used three types of background subtraction algorithm .The algorithm behind each method and the advantages as well as the disadvantages of these methods are also specified.

3.1 Background subtraction by Thresholding

A.The flowchart of the steps followed is given below



B. This example uses *VideoReader* (MATLAB), *implay*, and other Image Processing Toolbox functions to detect light-colored cars in a video of traffic.

Step 1: Access the Video

The VideoReader function constructs a multimedia reader object that can read video data from a multimedia file.

Then the get method is used to get the general properties and the setting of the video file such as frame rate per second.

Step 2: **Explore the video**

Then the video is played using the *implay* function.

This is a single frame from the input video.

Step 3: **Convert the image into grayscale**

To simplify the processing the RGB video is converted into it's corresponding grayscale version using *rgb2gray* function.

The video we have considered contains both light and dark colored cars.

Step 4 : **Develop the Algorithm**

While working with a video file it is better to apply the process or algorithm to a single frame and repeat the steps till all the frames in the video is processed .This is done using a *while* loop. For this car-tagging application take a frame that includes both light-colored and dark-colored cars. When an image has many structures, like the traffic video frames, it is useful to simplify the image as much as possible before trying to detect an object of interest. One of way is to do this is to suppress all objects in the image that are not light-colored cars (dark-colored cars, lanes etc.). Typically, it takes a combination of techniques to remove these extraneous objects.

One way to remove the dark-colored cars from the video frames is to use the *imextendedmax* function. This function returns a binary image that identifies regions with intensity values above a specified threshold, called regional maxima. All other objects in the image with pixel values below this threshold become the background. To eliminate the dark-colored cars, determine the average pixel value for these objects in the image. The pixel values can be extracted using the *implay* function . For this demo we have set the value to 50.

In the processed image, note that most of the dark-colored car objects are removed but many other extraneous objects remain, particularly the lane-markings. The regional maxima processing will not remove the lane markings since their pixel values are above the threshold. To remove these objects, we use the morphological function *imopen*. This function uses morphological processing to remove small objects from a binary image while preserving the large objects. When using morphological processing , the size and shape of the structuring element used in the operation should be decided first.Because the lane-markings are long and thin objects, we can use a disk-shaped structuring element with radius corresponding to the width of the lane markings. Here the *pixel region tool* is used in *implay* to estimate the width of these objects. For this demo the value is set to 2.

Step 4**: Apply the Algorithm to the Video**

This application processes the video one frame at a time in a loop .As a typical video contains a large number of frames, it would take a lot of memory to read and process all the frames at once.For faster processing the memory is preallocated here.

After a particular vehicle is detected the centroid of it is calculated and it is marked by a red colored square that represents the moving vehicle.

Step 5 :**Visualize the result**

The result can be viewed using the *implay* function.

C .Advantages of this method:

- simple to understand
- computation method is simpler
- uses a very less amount of memory

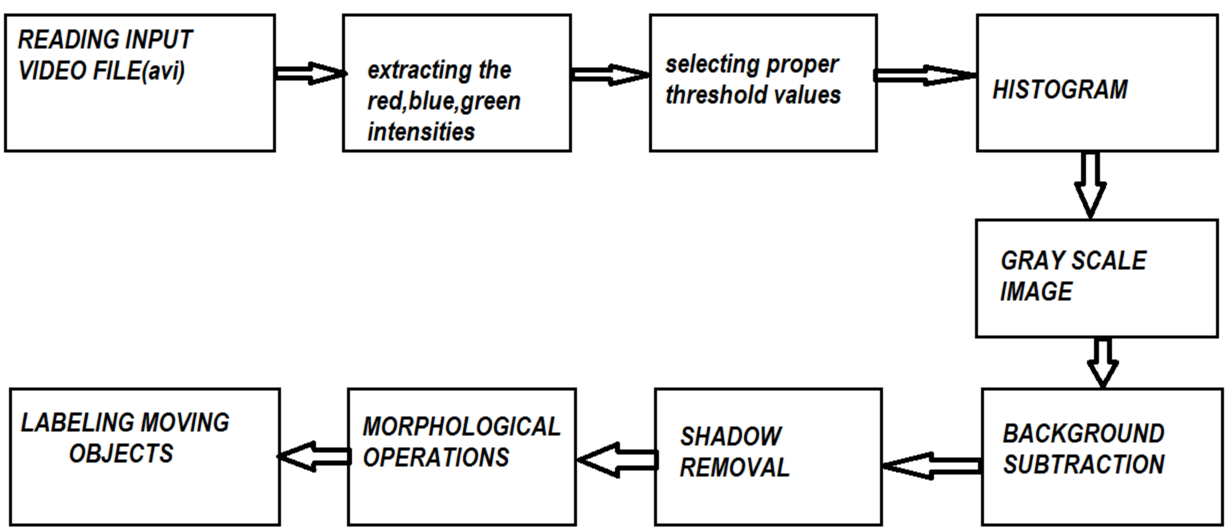
D .Disadvantages :

- dark-colored cars are rejected along with the background and only light-colored cars are detected

- sometimes white colored cars can be counted twice due to the separation by the dark colored glass.

3.2 Background subtraction by frame differencing:

A . Block diagram for the implemented algorithm



B. Description

Step 1 .Input Video

The input video format is avi. avi stands for audio video interleave. An AVI file actually stores audio and video data under the RIFF (Resource Interchange File Format) container format. In AVI files, audio data and video data are stored next to each other to allow synchronous audio with- video playback. Audio data is usually stored in AVI files in uncompressed PCM (Pulse-Code Modulation) format with various parameters. Video Data is usually stored in AVI files in compressed format with various codecs and parameters. The *aviread*, *aviinfo* functions are mentioned to read the input video avi format.

Step 2. Extraction:

After reading the input video file, extracted the red, green and blue intensities separately to find out the histogram easily. Image(:,:,1),image(:,:,2) and image(:,:,3) functions are used to read the red, blue and green intensities of input video frames.

Step 3. Threshold Values

Proper threshold values have to be chosen for background, standard deviation and area of the moving objects. The statistical parameter standard deviation is used in the processing of removing the shadow of the moving object. In this algorithm threshold value of background chosen as 250 pixels, standard deviation is 0.25 and area of the moving object is 8 pixels.8*8 pixel is taken as one block in this algorithm.

Step 4.Grayscale image:

Grayscale images are images without color, or achromatic images. The levels of a gray scale range from 0 (black) to 1 (white). After calculating the histogram, images are converted in to gray scale image using *rgb2gray* function in MATLAB.

Step 5.Subtraction

This proposed algorithm dynamically extracting the background from incoming all video frames, it is subtracted from every subsequent frame and compared with the background threshold. If is greater than the background threshold, it assumed as foreground otherwise it is background. The Background is updated in each and every frame.

$$B_t=(1-a)\, B_{t-1}+ a\, I_c$$

where Bt is the color vector of the background model in the t frame, I is the actual color vector of the same pixel in the frame t and a is the coefficient that declares the rate of adaptation with values in the range 0–1.

Step 6. **Shadow removal**

Performing the operation using a function on each frame by 8*8 block wise and result is compared with the variance threshold. If the result is less than the variance threshold, it assumes as shadow and it takes logic 0 otherwise it takes logic 1.

Step 7. Morphological operations

Morphology is a broad set of image processing operations that process images based on shapes. Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. By choosing the size and shape of the neighborhood, you can construct a morphological operation that is sensitive to specific shapes in the input image.

C .Advantages:

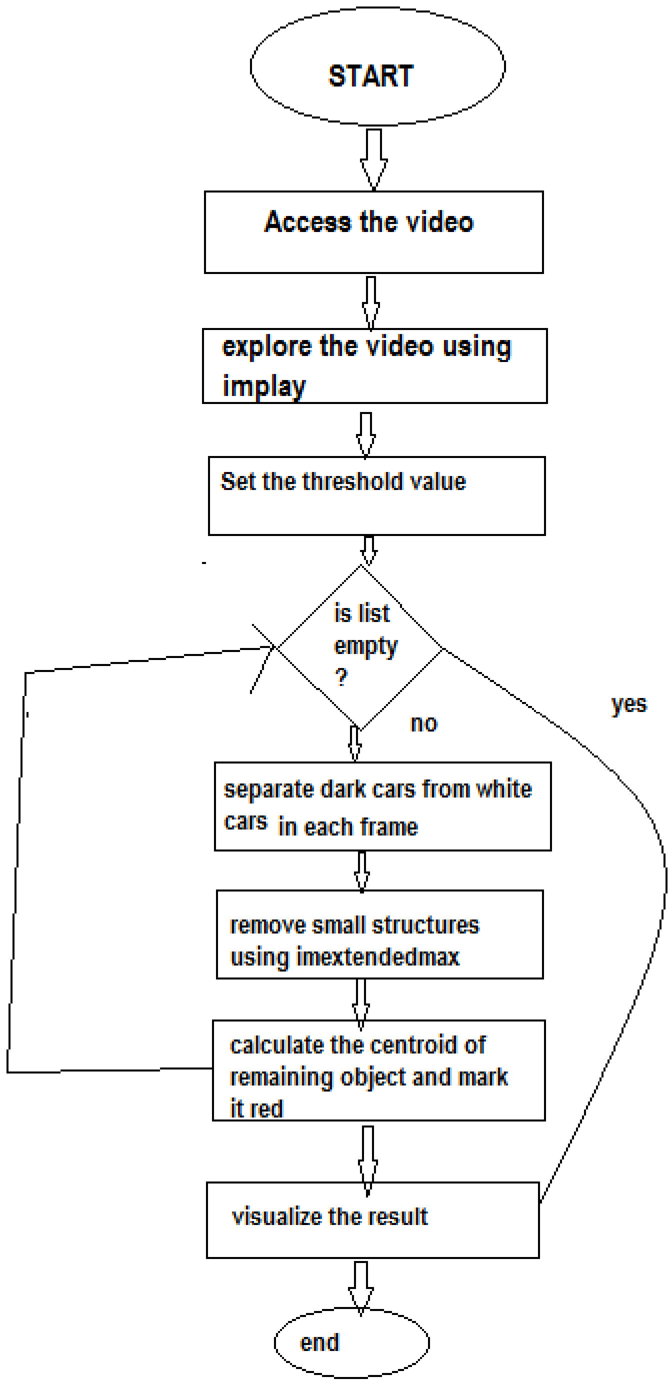
Since background is updated each time and not a general threshold value is used this method is more efficient than the previous one.

D .Disadvantages

As this method comprises the step of decomposing the video into frames and than compressing them again it involves a lot of memory to store the frames.Besides it involves complex calculations in every step for determining the foreground.

3.3 .Background Subtraction Using Statistical Method(Gaussian Mixture Model)

A .Flowchart of the algorithm implemented



B .Description:

Detecting and counting cars can be used to analyze traffic patterns. Detection is also a first step prior to performing more sophisticated tasks such as tracking or categorization of vehicles by their type.

This example shows how to use the foreground detector and blob analysis to detect and count cars in a video sequence. It assumes that the camera is stationary.

Step 1: Import the Video and Initialize Foreground Detector

Rather than immediately processing the entire video, first we obtain an initial video frame in which the moving objects are segmented from the background. This helps to gradually introduce the steps used to process the video.

The foreground detector requires a certain number of video frames in order to initialize the Gaussian mixture model. This example uses the first 50 frames to initialize three Gaussian modes in the mixture model.

Video Frame

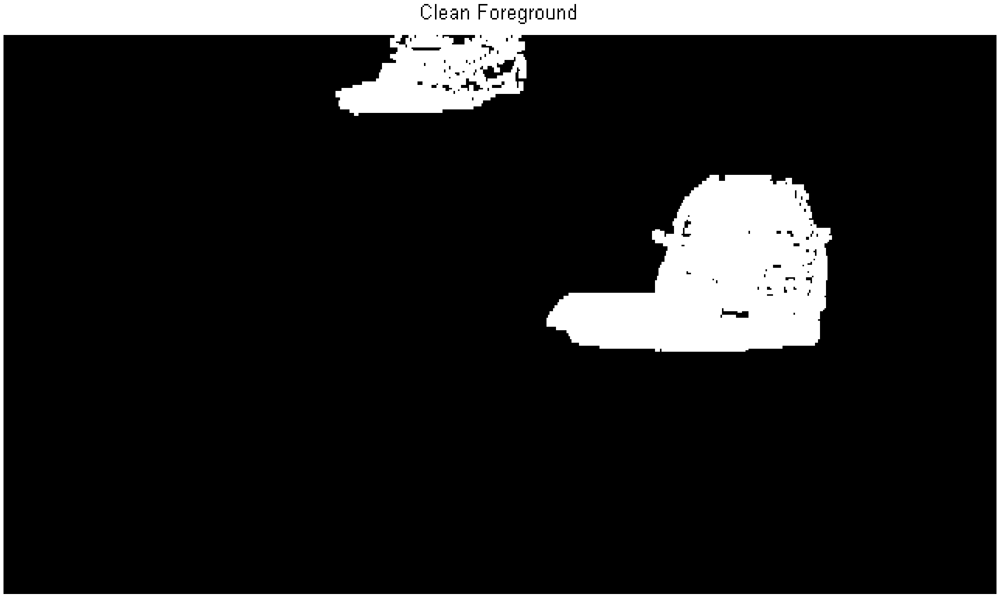
Foreground

Clearly the obtained image contains many noises in the background.To clean the backgroubd we follow the next step.

Step 2 :Remove noises

Morphological operations are used to remove the noises and get a cleaner foreground.This can be done using *imopen* function in MATLAB.

Clean Foreground



Step 4 :Find the bounding box

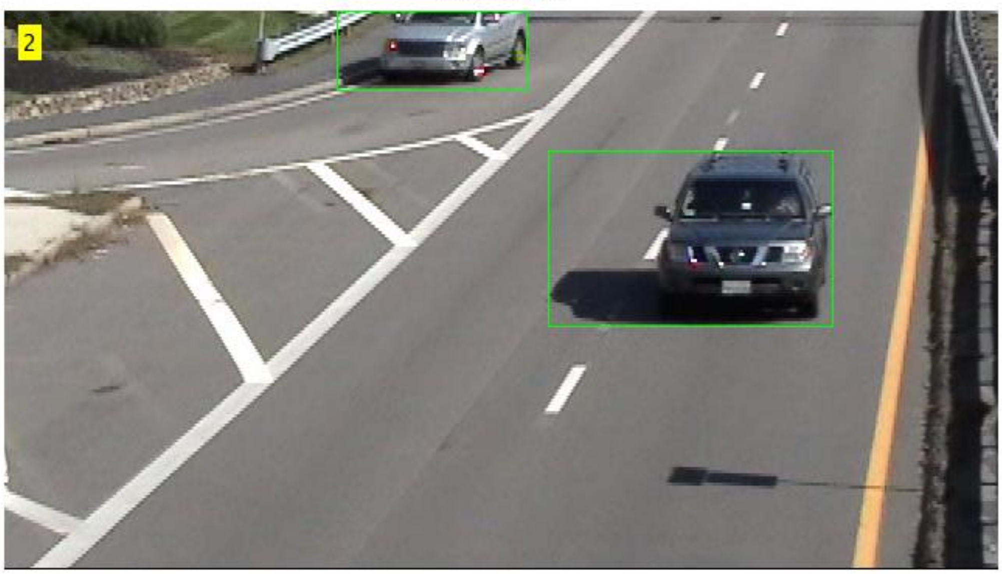
We find bounding boxes of each connected component corresponding to a moving car by using *vision.BlobAnalysis* object. The object further filters the detected foreground by rejecting blobs which contain fewer than 150 pixels

```
blobAnalysis    =    vision.BlobAnalysis('BoundingBoxOutputPort', true,  ...'AreaOutputPort',  false,  'CentroidOutputPort',  false,
...'MinimumBlobArea', 150);
bbox = step(blobAnalysis, filteredForeground);
```

Step 5:Insert rectangle around the moving vehicle:

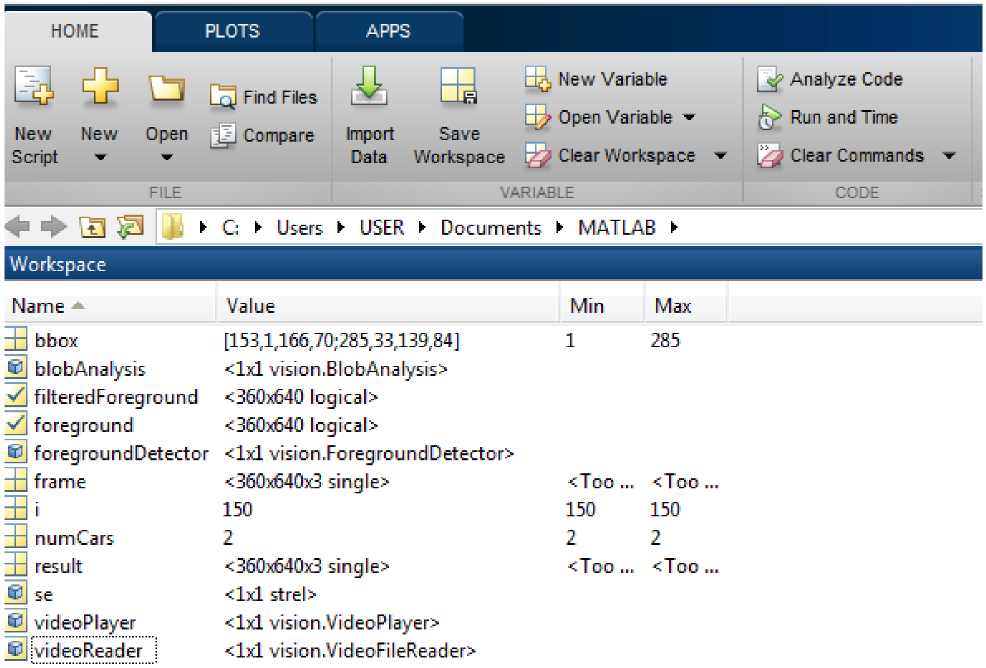
The moving vehicle once detected we calculate the centroid and size of the bounding box.Then we use a *insertshape* function to draw a rectangle around it.

Detected Cars



Step 6:calculate the number of vehicles

The number of vehicles in the current frame can be calculated from the number of bounding boxes .This method enables us to have a general idea about the vehicle frequency.



Step 6:Process the whole video

We use a *while* loop to process the frames till whole video is processed .Here we use a *release(videoPlayer*) function to allow morphological operations on the video.

C .Advantages

- This method uses a training set to determine the foreground, hence it is more efficient in giving a perfect output.
- Since generalized threshold is not used,it is able to detect white cars as well as the dark colored cars which was not possible using a simple thresholding function
- The number of vehicles in the current frame can be calculated .As a result vehicle frequency can be calculated that is helpful in avoiding traffic congestion.

D .Disadvantages

Though this method is more efficient than other methods it has other disadvantages such as shadow is not removed and it can confuse the user of the size of the vehicles.

4. Library functions used in implementation of above algorithms

- **Videoreader**

VideoReader Create a multimedia reader object.

OB = VideoReader(FILENAME) constructs a multimedia reader object, OB, that can read the video data from a multimedia file. FILENAME is a string specifying the name of that multimedia file. There are no restrictions on file extensions since by 26 default, MATLAB looks for the file FILENAME on the MATLAB path. If the object can't be constructed for any reason (e.g if the file can't be opened or does not exist, or if the file format is not recognized or supported), then MATLAB throws an error.

- **Implay**

implay Play videos, movies, or image sequences.

implay opens a movie player for showing MATLAB videos, movies or image sequences . The implay File menu select the movie or image sequence that you want to play. We can use implay controls to play the movie, jump to a specific frame in the sequence, change the frame rate of the display, or perform other exploration activities. You can open multiple implay movie players to view different movies simultaneous implay('filename') opens the implay movie player, displaying the content of the file specified by the given name. implay reads one frame at a time, conserving memory during playback. *implay* does not play audio tracks.

- **rgb2gray**

rgb2gray Convert RGB image or colormap to grayscale.

rgb2gray converts RGB images to grayscale by eliminating the hue and saturation information while retaining the luminance.

I = rgb2gray(RGB) converts the truecolor image RGB to the grayscale intensity image I.

- **imopen**

imopen Morphologically open image.

IM2 = imopen(IM,SE) performs morphological opening on the grayscale or binary image IM with the structuring element SE. SE must be a single structuring element object, as opposed to an array of objects.

- **Imextendedmax**

imextendedmax is an Extended-maxima transform.

BWM = imextendedmax(I,H) computes the extended-maxima transform, which is the regional maxima of the H-maxima transform where H is a nonnegative scalar.

Regional maxima are connected components of pixels with the same intensity value, t, whose exterior boundary pixels all have a value less than t.

- **vision.VideoPlayer**

VideoPlayer Play video or exhibit image

videoPlayer = vision.VideoPlayer returns a video player System object, videoPlayer, for displaying video frames. Each call to the step() method, displays the next video frame.

videoPlayer = vision.VideoPlayer('Name', 'Value') configures the video player properties, specified as one or more name-value pair arguments.

- **vision.ForegroundDetector**

ForegroundDetector Detects the foreground using Gaussian Mixture Models

H = vision.ForegroundDetector proceeds a foreground detector System object,H, that computes foreground mask using Gaussian Mixture

Models (GMM) given a series of either grayscale or color video frames.

H = vision.ForegroundDetector('PropertyName', PropertyValue, ...)

returns a foreground detector System object, H, with each specified

property set to the specified value.

Where *NumTrainingFrames* denotes the Number of initial video frames used for training the background model and *LearningRate* denotes the Learning rate used for parameter updates

- **vision.BlobAnalysis**

BlobAnalysis gives Properties of connected regions

HBLOB = vision.BlobAnalysis returns a blob analysis System object,

HBLOB, used to compute statistics for connected region in a binary

image.

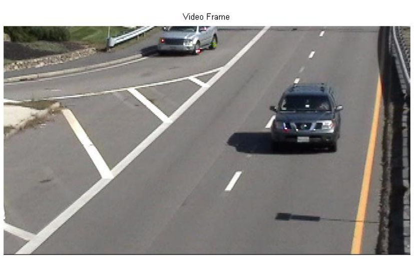
5 .Results

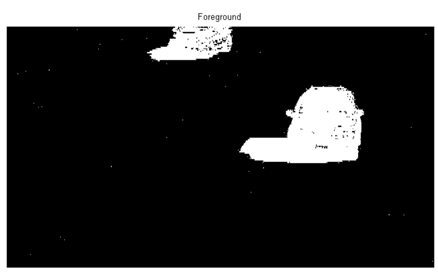
Here we represent the comparative result of all the algorithms implemented.

Method 1:Using simple thresholding function result into a simpler calculation but rather not desirable results as it may contain the vehicles having grayvalue more than a certain value.

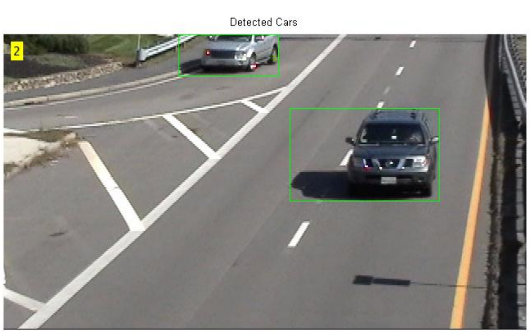
Method 2:Using frame differencing method will give a more desirable result than the previous one.But since we first divide the video into a number of frames and than compress them together again to make the output video it involves a lot more complex calculation and memory.

Method 3:Statistical method or here Gaussian mixture model uses 50 number of training frames to initialize 3 Gaussian modes.Here the parameters like mean and variance used are not arbitrary and output is more desirable.





The final output can be represented by a rectangle around it and the frequency of cars is calculated based on it.



6. Conclusion

We presented a vehicle detection and tracking system based on statistical model of vehicle detection. Object detection in videos obtained from stationary camera and fixed background is achieved through *background subtraction* approach. In this approach a background model is developed considering the first frame or first few frames as training set. Consequently, a thresholding technique is utilized to extract foreground objects. Shadows are often misclassified as foreground objects, which needs an additional step i.e morphological operation to remove noises before the detected objects can be tracked. Object tracks are computed by various approaches. Centroid in subsequent frames are searched in a fixed size window, which makes the algorithm more complex.

For the last two decades, researchers across the world have been working towards object detection and tracking as well. Significant volumes of literature are available in this field. Real time employment of the algorithm demands higher accuracy with less complexity, which makes the problem still open and needs significant study. In this thesis, efforts have been made to detect vehicle in an input video file.

In this chapter three the algorithms are presented. The first method involves a thresholding method where simply the values below a certain level are rejected .In second method comparision is done frame by frame to get the foreground .While in third method a Gaussian mixture model is used which uses a set of training frames to determine the foreground .Consequently the average vehicle frequency is calculated from the number of moving vehicles per frame.This result is very useful in prevention of traffic congestion.

Scope for future work:

In future work the proposed algorithm can be extended for night surveillance, where some primary tests leave space for improvement on the existing algorithms reported in literature. However, the other modules of the proposed system should be improved, focusing on the occlusion handling and vehicle matching procedure. Moreover, it remains a challenge to utilize the capabilities of the proposed algorithm to other kind of machine vision problems, such as security, remote sensing, ship surveillance and a plethora of surveillance applications.

6. References

1.R C Gonzalez, Richard E woods. *Digital image processing*, Pearson education ,United States,3rd edition

2. Nicholas A. Mandellos , Iphigenia Keramitsoglou , Chris T. Kiranoudis *A background subtraction algorithm for detecting and tracking vehicles.* Department of Process Analysis and Systems Design, National Technical University of Athens, GR-15780 Athens, Greece

3. Kinjal A Joshi, Darshak G. Thakore, *A Survey on Moving Object Detection and Tracking in Video Surveillance System,* International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-2, Issue-3, July 2012

4. Chin-Teng Lin1, Sheng-Chih Hsu1, *Boosted Vehicle Detection Using Local and Global Features* ,Institue of Electrical and Control Engineering, College of Electrical and Computer Engineering of the National Chiao-Tung Univer-sity, Hsinchu, Received June 16th, 2013; revised July 18th, 2013; accepted July 26th, 2013

5. Gottipati. Srinivas Babu *Moving Object Detection Using MATLAB,* International Journal of Engineering Research & Technology (IJERT) ,Vol. 1 Issue 6, August – 2012

6. YK Wang, SH Chen. Robust vehicle detection approach[C]. IEEE Conference on Advanced Video and Signal Based Surveillance, IEEE Press, 2005, pp. 117-122.

7. Muzaffar Djalalov1, Humaira Nisar, Yasir Salih2, Aamir Saeed Malik2, *Member*, *IEEE An Algorithm For Vehicle Detection And Tracking Universiti Teknologi* ,PETRONAS in 2009-2010.,

8. Gupte, S., O. Masoud, R. F. K. Martin, and N. P. Papanikolopoulos. "*Detection and Classification of Vehicles",* IEEE Transactions on Intelligent Transportation Systems, Vol. 3, No. 1, March 2002, pp. 37-47

9. K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: Principles and practice of background maintenance. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 1, pages 255 – 261, Sep 1999.